LISTING OF CLAIMS

The listing of claims provided below replaces all prior versions, and listings, of claims in the application.

1. (Currently Amended) A processor for executing a secure hash algorithm (SHA) computation on a message, comprising:

a core having a first execution unit and a second execution unit, wherein an output of the first execution unit is connected to an input of the second execution unit,

wherein the first execution unit is defined to perform a message schedule computation on a data block of the message to generate an expanded representation of the data block from a first number of bits to a larger number of bits,

wherein the first execution unit is defined to communicate a partial result of the expanded version of the data block ~~schedule computation on the data block~~ through its output to the input of the second execution unit when the partial result becomes available and prior to completion of the message schedule computation on the data block,

wherein the second execution unit is defined to perform a compression function on the partial result received from the first execution unit in parallel with the first execution unit continuing the message schedule computation on the data block, whereby the compression function is defined to iteratively consume the partial result.

2. (Previously Presented) A processor for executing a secure hash algorithm (SHA) of claim 1, wherein the first execution unit is a single instruction multiple data (SIMD) execution unit.

3. (Previously Presented) A processor for executing a secure hash algorithm (SHA) of claim 1, wherein the second execution unit is an integer execution unit.

4. (Previously Presented) A processor for executing a secure hash algorithm (SHA) of claim 1, wherein the message is a parsed padded message.

5. (Previously Presented) A processor for executing a secure hash algorithm (SHA) of claim 4, wherein the parsed padded message includes an original message and a plurality of pad bits, the original message being a plurality of bits.

6. (Previously Presented) A processor for executing a secure hash algorithm (SHA) of claim 1, wherein the partial result includes a group of bits represented as a hexadecimal value.

7. (Currently Amended) A processor for cryptographic computation, comprising:

a first execution unit defined to perform a message schedule computation on a data block and produce a partial result of the message schedule computation on the data block prior to completion of the message schedule computation on the data block, wherein the message schedule computation generates an expanded representation of the data block from a first number of bits to a larger number of bits, wherein the partial result includes a group of bits capable of being represented by a hexadecimal value, the first execution unit further defined to have an output through which the partial result is communicated; and

a second execution unit defined to have an input to which the output of the first execution unit is connected, the second execution unit defined to receive the partial result from the first execution unit through the input and to perform a compression function on the partial result while the first execution unit continues performing the message schedule computation on the data block, whereby the compression function is defined to iteratively consume the partial results.

8. (Previously Presented) A processor for cryptographic computation of claim 7, wherein the first execution unit is defined to receive a plurality of blocks, the plurality of blocks including an original message and a plurality of pad bits.

9. (Previously Presented) A processor for cryptographic computation of claim 8, wherein the first execution unit is defined to perform a rotation operation on the plurality of blocks as part of the message schedule computation.

10-11. (Cancelled)

12. (Currently Amended) A method, comprising:

receiving a message; and

performing a cryptographic computation on the message, the cryptographic computation including a hash computation including,

performing a message schedule computation on a block of data using a first execution unit, wherein the message schedule computation generates an expanded representation of the data block from a first number of bits to a larger

number of bits, whereby a partial result of the message schedule computation is generated prior to completion of the message schedule computation,

communicating the partial result from an output of the first execution unit to an input of a second execution unit while the message schedule computation on the block of data continues using the first execution unit, and

performing a compression function on the partial result using the second execution unit while the message schedule computation on the block of data continues using the first execution unit, whereby the compression function is defined to iteratively consume the partial result.

13. (Previously Presented) A method of claim 12, wherein the cryptographic computation includes a preprocessing operation including,

padding the message to generate a padded version of the message;

parsing the padded version of the message; and

setting initial hash values to be used in the hash computation.

14. (Cancelled)

15. (Original) A method of claim 12, wherein performing the message schedule computation further includes assigning rotated bits in the block of data to the partial result.

16. (Cancelled)

17. (Currently Amended) A method for a one-way cryptographic hash computation, comprising:

operating a first execution unit to perform a message schedule computation on a data block to produce a partial result of the message schedule computation on the data block, wherein the schedule computation generates an expanded representation of the data block from a first number of bits to a larger number of bits;

sending the partial result through an output of the first execution unit to an input of a second execution unit while the first execution unit continues to operate to perform the message schedule computation on the block of data; and

operating a second execution unit to perform a compression function on the partial result while the first execution unit continues performing the message schedule computation on the data block, whereby the compression function is defined to iteratively consume the partial result.

18. (Previously Presented) A method for a one-way cryptographic hash computation of claim 17, wherein operating the first execution unit to perform the message schedule computation includes rotating bits in the data block.

19. (Previously Presented) A method for a one-way cryptographic hash computation of claim 17, wherein operating the second execution unit to perform the compression function includes rotating bits in the partial result.

20-27. (Cancelled)